A tree does not make a well-formed sentence: Improving syntactic string-to-tree statistical machine translation with more linguistic knowledge

Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 32 (2015) 27–45

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

A tree does not make a well-formed sentence: Improving syntactic string-to-tree statistical machine translation with more linguistic knowledge☆

Rico Sennrich*, Philip Williams, Matthias Huck

School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, Scotland, UK

Received 1 March 2014; received in revised form 24 June 2014; accepted 6 September 2014
Available online 16 September 2014

Abstract

Synchronous context-free grammars (SCFGs) can be learned from parallel texts that are annotated with target-side syntax, and can produce translations by building target-side syntactic trees from source strings. Ideally, producing syntactic trees would entail that the translation is grammatically well-formed, but in reality, this is often not the case. Focusing on translation into German, we discuss various ways in which string-to-tree translation models over- or undergeneralise. We show how these problems can be addressed by choosing a suitable parser and modifying its output, by introducing linguistic constraints that enforce morphological agreement and constrain subcategorisation, and by modelling the productive generation of German compounds.

Keywords: Statistical machine translation; Syntactic translation models; String-to-tree models; Morphology

1. Introduction

The modelling limitations of phrase-based statistical machine translation (SMT) are well known, for instance its inability to model discontiguous phenomena such as verb complexes in German, and the limitation to local fluency modelling. Hierarchical models and synchronous context-free grammars (SCFGs) are an attractive alternative because they do not suffer from these theoretical limitations. We can learn an SCFG from a parallel corpus that is syntactically annotated. For a string-to-tree system, annotation of the target side is sufficient. During decoding, such an SCFG is used to build a target-side syntactic tree from a source string. A common expectation might be that building a syntactic tree ensures that the sentence that is produced is grammatically well-formed, but in reality, this is often not the case.

In this work, we discuss why string-to-tree SMT systems can produce ungrammatical output, and we examine the reasons in detail. Specifically, we investigate the following crucial aspects:

☆ This paper has been recommended for acceptance by R.K. Moore.

* Corresponding author. Tel.: +44 7847842380.
E-mail addresses: v1rsennr@staffmail.ed.ac.uk, rico.sennrich@gmx.ch (R. Sennrich), P.J.Williams-2@sms.ed.ac.uk (P. Williams), mhuck@inf.ed.ac.uk (M. Huck).

http://dx.doi.org/10.1016/j.csl.2014.09.002

- Data sparseness issues, specifically unknown words.
- Overgeneralisation phenomena of SCFG models.
- The relevance of the syntactic annotation scheme for specific linguistic phenomena.
- The impact of morphosyntactic ambiguities.
- Problems related to productive compositional morphology.

We describe the inclusion of linguistic features into the translation process to promote grammatical translation output, including a unification-based morphological agreement checks for noun phrases, subcategorisation constraints for verbs, and a target-side compound splitting and merging approach that makes use of a finite-state morphology.

2. String-to-tree translation models

In most modern syntax-based SMT models, the translation units are either SCFG rules or synchronous tree-substitution grammar (STSG) rules.[1] For the purposes of string-to-tree decoding, the two formalisms are equivalent unless the decoding model uses the internal structure of the STSG rules to define scoring features. Since SCFG is more widely used in the literature, we will use SCFG throughout this paper, but note that all points apply equally to STSG.

2.1. Synchronous context-free grammar

In its most general form, a SCFG rule is a rewrite rule:

$$\langle A, B\rangle \rightarrow \langle \alpha, \beta, \sim\rangle$$

where the head is a pair of source and target non-terminals, A and B, and the body comprises a string, α, of source terminals and non-terminals; a string, β, of target terminals and non-terminals; and a one-to-one correspondence $\sim$ between source and target non-terminals. As in context-free grammar, the terminals and non-terminals are atomic symbols.

As the name SCFG implies, derivation using an SCFG involves the same assumption of context-freeness as in CFG: a pair of linked source and target non-terminals, A and B, in a sentential form can be rewritten using the body of some rule, r, provided that that r's non-terminal head symbols match. For the purposes of bottom-up parsing, a synchronous subderivation with head non-terminals A and B is equivalent to any other with the same head symbols.

2.2. String-to-tree grammars

In string-to-tree models, only one non-terminal symbol, X, is used on the source side of the grammar. As in hierarchical phrase-based SMT (Chiang, 2005, 2007), the X non-terminal is used generically to represent a gap in a discontiguous phrase (here we use "phrase" in the same sense as in phrase-based SMT: a sequence of words). In contrast, the vocabulary of target non-terminal symbols may be arbitrarily large. Depending on the grammar learning approach, it may comprise tens, hundreds, or even thousands of distinct symbols. Typically, these are derived from the constituent labels of phrase-structure parse trees. In the following rules:

$$\langle \mathrm{X}, \mathrm{NP}\rangle \rightarrow \langle \textit{the dog}\,,\, \textit{der Hund}\,\rangle$$
$$\langle \mathrm{X}, \mathrm{SENT}\rangle \rightarrow \langle \textit{Then}\,\mathrm{X}_1\,\textit{barked}\,,\, \textit{Dann bellte}\,\mathrm{NP}_1\,\rangle$$

the head non-terminals are used to label the string *der Hund* as an NP and the string *Dann bellte* NP_1 as a SENT. (In the rule body, the subscripts are used to indicate the non-terminal correspondence.) One or more additional non-terminal symbols are used for the "glue" rules, which concatenate partial derivations.

There are two main approaches to rule extraction for string-to-tree models: the first extends Chiang (2005)'s SCFG extraction method to incorporate target-side annotation derived from the labels of phrase-structure parse trees. This is the approach first described in the syntax-augmented MT (SAMT) model (Venugopal and Zollmann, 2006). The

[1] STSG is a variant of synchronous tree-adjoining grammar (Shieber and Schabes, 1990) that includes the substitution operation but not the adjunction operation.

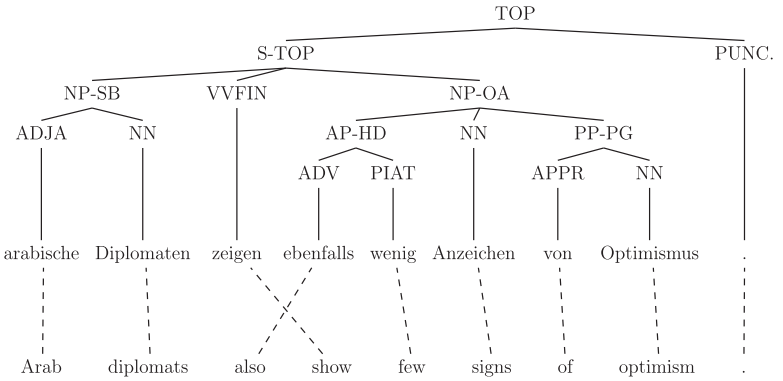

Fig. 1. A word-aligned sentence pair annotated with a target-side parse tree.

second is GHKM (Galley et al., 2004, 2006), which derives STSG rules from training data annotated in the same way. The two approaches are closely related (Hanneman et al., 2011; Hopkins et al., 2011) and differ in details such as the restrictions they place on extracted rule size, the handling of unaligned words, and the requirement that the target side of the extraction site is covered by a parse-tree constituent.

The SAMT and GHKM rule extraction algorithms are dependent on automatic word alignments. Fig. 1 shows a word-aligned sentence pair, annotated with a phrase-structure parse tree on the target side. Both algorithms employ phrase-based style heuristics that require a rule extraction site to contain consistent word alignments. For instance, an extraction site that covers the source span *also show few* must include the aligned words *zeigen ebenfalls wenig*. Given the input sentence pair of Fig. 1, it is not possible to extract, for example, the rule

$$\langle \mathrm{X}, \mathrm{AP-HD} \rangle \rightarrow \langle \textit{also show few}\,,\, \textit{ebenfalls wenig} \rangle$$

because it is not consistent with word alignment, nor is it possible to extract a rule spanning *zeigen ebenfalls wenig* because the phrase is not covered by a constituent.

2.3. Decoding

Decoding in an SCFG-based model involves searching the space of synchronous derivations for the highest-scoring translation, according to some scoring model. A single translation can have many possible derivations and for decoding to be computationally tractable, the search criterion is typically approximated by a search for the highest-scoring derivation; the search itself is typically an approximate beam search.

As in phrase-based SMT, derivations are usually scored according to a log-linear model (Och and Ney, 2002) that allows for the incorporation of arbitrary feature functions defined over the source string and target derivation. To facilitate efficient dynamic programming, the feature functions should be defined such that they are local to SCFG rules in order that the score is decomposable along subderivation boundaries. In practice, the n-gram language model, which violates this desideratum, is sufficiently important for translation quality that it is integrated at the expense of search efficiency. n-gram language model integration is usually achieved using cube pruning (Chiang, 2007).

Typically, the feature functions of a string-to-tree model include scores for the individual rules such as the bidirectional translation probabilities $p(\alpha \mid \beta,\ B)$ and $p(\beta,\ B \mid \alpha)$; bidirectional lexical translation probabilities (Koehn et al., 2003); the number of terminals in β; a constant rule penalty; and some measure of rule frequency. Marcu et al. (2006) and Williams and Koehn (2012) both use an unlexicalised PCFG grammar to score the tree fragment from which a rule is extracted. This feature is intended to encourage the production of syntactically well-formed derivations, and essentially serves as a syntactic language model.

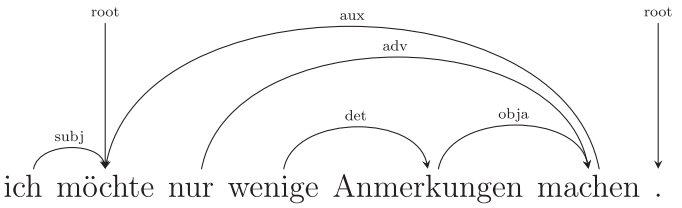

Fig. 2. A ParZu dependency tree.

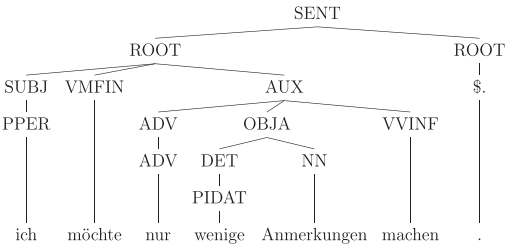

Fig. 3. Constituency representation of the ParZu dependency tree.

3. An English→German string-to-tree model

We will focus our discussion and experiments on the translation direction English→German. As background to our linguistically motivated extensions of a string-to-tree SMT system, we will first discuss the linguistic annotation of the German target text, and the role of glue rules and non-terminal labels for unknown words.

3.1. Syntactic annotation of German

The syntactic annotation of the target text in a parallel training corpus has various effects for string-to-tree translation modelling. Non-terminal symbols constrain which rule rewrites are allowed during parsing, and the size of its vocabulary is a trade-off between sparseness (if the vocabulary is large) and overgenerality (if it is small). The degree of branching of syntactic trees affects rule extraction, where only aligned phrases that cover a constituent are extracted as rules, unless we relax this constraint as in the SAMT model. Parsing errors, in particular systematic ones, result in ungrammatical patterns being learned by our SCFG. While annotation schemes of German treebanks and their impact on parsers has been discussed in the parsing literature (Kübler, 2005), our aims in this work is to adapt the output of a syntactic parser to fit the need of the downstream application, which is machine translation.

We focus on the annotation scheme used by ParZu (Sennrich et al., 2013). ParZu is a syntactic parser which implements the dependency grammar described in (Foth, 2005) and is trained on the dependency representation of the TüBa-D/Z treebank (Telljohann et al., 2004; Versley, 2005). Part-of-speech tags from the Stuttgart-Tübingen tagset for German (STTS) (Schiller et al., 1999) are used as pre-terminal labels. Parse trees may be non-projective, but since the SCFG model requires a context-free annotation we use the projective representation which is optionally provided by ParZu. We convert the dependency trees into a constituency representation by considering each token to be the head of a constituent, using its dependency label as non-terminal symbol, and adding a virtual root node SENT, to which all words without a head (this typically includes the finite verb of main clauses, sentence-final punctuation, and sentence fragments that were left unattached) are attached. Fig. 2 shows a dependency tree produced by ParZu, and Fig. 3 shows the same tree in a constituency representation.

3.2. Glue grammar and unknown word labels

The most obvious reason for ill-formed translations is when parsing fails to produce a full tree. SCFG systems typically resort to a concatenation of partial trees in this case, implemented through a set of glue rules.

The root cause for an inability to form complete trees is often data sparseness, in particular words that are unknown to the SCFG. However, when discriminatively optimising the cost of the glue rules on a development set, the system

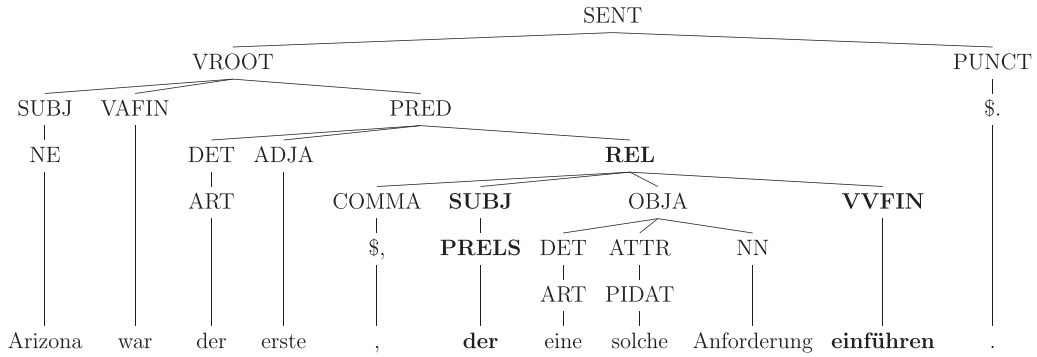

Fig. 4. Translation output with lacking subject–verb agreement.

may learn to use them more frequently. It is also common to limit the maximum span of CFG trees for efficiency reasons. With Scope-3 pruning (Hopkins and Langmead, 2010), the complexity of parsing is cubic to the input length, albeit with a high constant due to grammar size. Nadejde et al. (2013) use a maximum span of 25 for the string-to-tree grammar.

To allow the production of syntactic trees even if words in the source string are unknown to the SCFG, Nadejde et al. (2013) use statistical evidence from the training data, specifically the label distribution of singletons, to assign probabilities to different non-terminal labels for unknown words. Assigning a suitable non-terminal label to unknown words is important because its label constrains the possible derivations of the sentence.

As an alternative strategy, we propose to instead use sparse features to discriminatively learn which labels to use for unknown words during decoding. We initially label unknown words with UNK, and relax the matching constraint during rule application. Instead of requiring each non-terminal symbol in the body of a rule to exactly match the head of the rule that is substituted into it, we also allow a number of soft matches. Specifically, we allow soft matches from UNK to all other non-terminal symbols, and trigger a sparse feature for every soft (and exact) match that identifies the two non-terminal symbols of the rule expansion. Also, since our syntactic constraints that we discuss later rely on the internal tree structure, we want to avoid the use of glue rules, and thus fix their cost at a sufficiently high value so that they are only used if no other derivation can be found, and set the maximum span of rules to 50.

4. Overgeneralisations in a SCFG model

It is easily apparent why translations that are a product of incomplete derivations fail to be syntactically correct. However, even full trees that are produced by the SCFG translation model, and deemed acceptable by both the *n*-gram language model and the target-side PCFG, may be ill-formed. This is the result of the independence assumptions of the model, which scores rules independently and treats rules with the same head symbol as interchangeable, and overgeneralisations in the linguistic annotation. Overgeneralisations in the set of syntactic labels are typically unproblematic for parsing because modern parsers use a rich feature set and do not make such strong independence assumptions as the SCFG and target-side PCFG that we use for decoding. Also, the ability to discriminate between grammatically correct and incorrect sentences is not a central goal for most probabilistic parsers, with the main evaluation criterion being performance on natural text (e.g. Nivre et al., 2007; Kübler, 2008). While it would be desirable to use syntactic parsers to distinguish between well-formed and ill-formed translation hypotheses, in past research on using parsers as language models, parser scores failed to improve translation, partially because the parsers used gave high scores to ungrammatical hypotheses (Och et al., 2004; Post and Gildea, 2008).

An example of a grammatical error that is produced by our baseline SCFG is shown in Fig. 4. In this example, subject–verb agreement is violated in the relative clause, the correct inflectional form being the 3rd person singular form *einführte* (Engl: introduced), instead of the plural form *einführen*. Since person and number are not encoded in

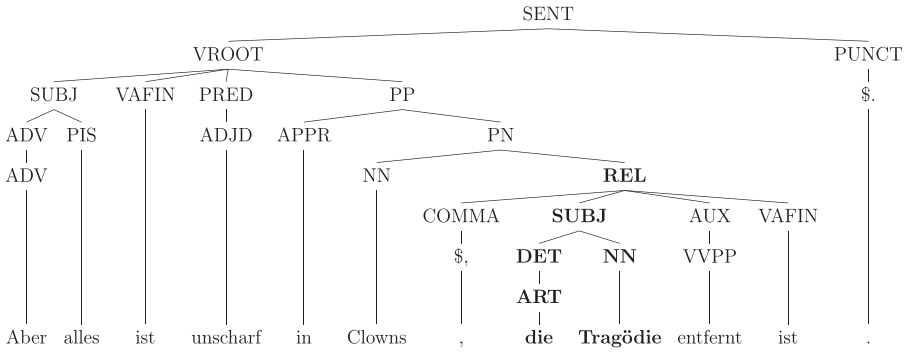

Fig. 5. Translation output with ill-formed relative clause.

the SUBJ symbol, nor in the pre-terminal symbols of either the pronoun or the verb, the SCFG learns various rules which allow wrong subject–verb combinations, for instance the following:

REL → , SUBJ OBJA *einführen*
REL → , PRELS OBJA *einführen*
REL → , *der* OBJA VVFIN
REL → , SUBJ OBJA VVFIN

In other words, the grammar incorrectly assumes independence between the subject and the verb, unless we use a rule in which both are lexicalised. It is also apparent that an n-gram language model is unlikely to promote agreement due to the distance between the subject and the verb.

Morphological agreement, either between the subject and the verb or within a noun phrase, is a frequent problem in a morphologically rich language such as German. However, there are other overgeneralisations that our grammar makes. For instance, consider Fig. 5, in which the word order of the second clause is wrong because it is analysed as a relative clause, which has verb-last word order in German, rather than a coordination of two main clauses with verb-second word order. The analysis as a relative clause is obviously wrong because the clause does not start with a relative pronoun, but with the nominal subject *die Tragödie* (Engl: the tragedy). However, the label SUBJ does not specify whether the subject is relative or not, and a rule of the form REL → , SUBJ AUX VAFIN is perfectly consistent with the training data.

Even though the errors in Figs. 4 and 5 affect different linguistic phenomena, namely morphological agreement and word order, they share the same root cause: the SCFG assumption that subderivations with the same head symbol are equivalent, and that decoding can be decomposed into rule-local feature functions. Depending on the language and syntactic annotation, other structures will be affected by this assumption.

Our aim is to reduce the number of errors stemming from the independence assumption made during SCFG decoding. One potential solution is to increase the granularity of the non-terminal symbol set to introduce new rule derivation constraints. However, increasing the granularity of the non-terminal set, e.g. by enriching the labels with morphological information, can impose too many restrictions on decoding, and prevent valid generalisations, especially in a language such as German which is syncretic, i.e. where multiple morphological analyses share the same word form. For example, we typically want to enforce case, number and gender agreement within noun phrases, but because adjective inflection does not depend on gender in the plural, naively enriching the label set of noun and noun attribute non-terminals with the full set of morphological information would also prevent correct derivations.

5. Linguistically informed improvements to a syntactic system

5.1. Modifying the syntactic label set

We only perform minimal modifications to the original ParZu label set. Firstly, its dependency grammar does not analyse brackets and punctuation marks, and gives them the label ROOT. The same label is used for the verbal root of

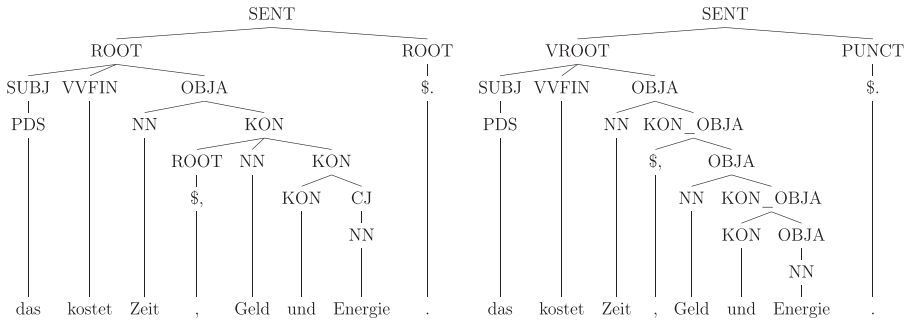

Fig. 6. Original ParZu representation of coordinations (left) and modified version that allows recursive rules (right).

a sentence, and any unattached structures that could not be fully parsed. We split the ROOT label in the treebank into five categories: brackets, commas, sentence-final punctuation marks (all easily identified by the pre-terminal labels), verbal roots of main clauses (VROOT) and other tree fragments (ROOT).

A second enrichment that we perform is concerned with coordinated elements, which are all given the label KON, or CJ for the last element, by ParZu. This is problematic because KON and CJ are overgeneral, being used for noun phrases, verb phrases, prepositional phrases, adverbs, and others. Instead, we copy the label of the coordination head to each conjoined element, and make the conjoined elements dependent on the preceding coordinating conjunction or comma, if they are not already. The label of subtrees headed by coordinating conjunctions is concatenated with the label of their head. This allows the model to learn generalisations such as:

$$\text{OBJA} \rightarrow \text{NN KON\_OBJA}$$

$$\text{KON\_OBJA} \rightarrow \textit{und}\ \text{OBJA}$$

The original and the modified annotation of a coordination are illustrated in Fig. 6.

Thirdly, we distinguish between prenominal and postnominal genitive modifiers. Since prenominal genitive modifiers are typically named entities, as in *Peters Vorschlag* (Engl: *Peter's proposal*), and postnominal ones noun phrases with an article, as in *der Vorschlag des Präsidenten* (Engl: *the proposal of the president*), separating the two types of genitive modifiers puts more constraints on word order during decoding.

5.2. Morphological agreement for noun phrases

German has a rich inflectional morphology, which marks grammatical features, like gender and case, on determiners, adjectives, noun, and verbs. Much of German's morphosyntax can be understood in terms of feature agreement and case government. For instance, the number and person features of a finite verb should agree with those of the subject; the case of a prepositional phrase is governed by the choice of preposition.

The production of inflection that is coordinated over multiple target words poses a problem since the words that bear the inflectional markers may be produced by the application of independent translation rules. Typically, the n-gram language model is the only means of enforcing consistency and this may be inadequate for longer-range agreement or for n-grams that were not seen during training.

Whilst morphological features could in principle be encoded in the non-terminal labels (for example, using SUBJ-SG-3-F to indicate a singular, third person, feminine subject), the use of highly specific labels runs the risk of exacerbating problems of training data sparsity. We therefore follow Williams and Koehn (2011) and use a unification-based approach to enforcing agreement. During training, we pass the target-side terminal vocabulary through the Zmorge morphological analyser (Sennrich and Kunz, 2014) and use the analyses to extract a lexicon of feature structures. The lexicon associates each target surface form with a set of feature structures. For example, two entries for the definite article *das* and the noun *Kätzchen* (Engl: kitten) are

$$das \rightarrow \begin{bmatrix} \text{CAT} & \texttt{ART} \\ \text{INFL} & \begin{bmatrix} \text{CASE} & \texttt{nom} \\ \text{DECLENSION} & \texttt{weak} \\ \text{AGR} & \begin{bmatrix} \text{GENDER} & \texttt{n} \\ \text{NUM} & \texttt{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad Kätzchen \rightarrow \begin{bmatrix} \text{CAT} & \texttt{NN} \\ \text{INFL} & \begin{bmatrix} \text{CASE} & \texttt{nom} \\ \text{AGR} & \begin{bmatrix} \text{GENDER} & \texttt{n} \\ \text{NUM} & \texttt{sg} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Syncretism is common in German and many target words (including *das* and *Kätzchen*) have multiple morphological analyses and therefore multiple entries in the lexicon with different feature values.

Our SCFG grammar rules are augmented with constraints: identities that require feature compatibility between feature structures. For example, in the following rule:

$$\begin{aligned} &\text{SUBJ} \rightarrow \textit{die}\ \text{ADJA}\ \text{NN} \\ &\langle \text{SUBJ}\ \text{INFL} \rangle = \langle \textit{die}\ \text{INFL} \rangle \\ &\langle \text{SUBJ}\ \text{INFL} \rangle = \langle \text{ADJA}\ \text{INFL} \rangle \\ &\langle \text{SUBJ}\ \text{INFL} \rangle = \langle \text{NN}\ \text{INFL} \rangle \\ &\langle \text{SUBJ}\ \text{INFL}\ \text{CASE} \rangle = \textit{nom} \\ &\langle \text{SUBJ}\ \text{INFL}\ \text{DECLENSION} \rangle = \textit{weak} \\ &\langle \textit{die}\ \text{CAT} \rangle = \textit{ART} \end{aligned}$$

the first three constraints ensure that the article *die* and the target words for the ADJA and NN subderivations have lexicon entries with inflection values that are compatible under unification. The fourth constraint requires nominative case. This constraint is based on the noun phrase label: subjects in German are indicated by the use of nominative case. The fifth constraint ensures that the inflection value of the ADJA is consistent with the weak declension paradigm (which is required for a noun phrase containing a definite article). The final constraint ensures that the lexicon entries considered for the terminal *die* are for the ART word class.

During decoding, a hypothesis's constraints are evaluated after it is popped from the cube pruning queue. If the constraints succeed then the hypothesis is added to the beam; otherwise the hypothesis is discarded.

We include constraints for the following phenomena:

1. Agreement of determiners and adjectives with the noun they modify.
2. Agreement of finite verbs with their subjects.
3. Choice of adjectival declension paradigm based on presence and definiteness of determiner.
4. Prepositional case government.
5. Selection of noun phrase case according to grammatical function.

Our constraint extraction algorithm is similar to that of Williams and Koehn (2011), but adapted for ParZu's parse tree style. It involves two steps: (i) the nodes of each training parse tree are grouped into sets according to their common membership of agreement and government relations; (ii) the GHKM rule extraction algorithm is extended to generate identities between terminals and non-terminals that belong to a common set.

5.3. Subcategorisation constraints

Additionally to noun phrase agreement, we model a number of subcategorisation phenomena relating to verbs and clauses. We do this through a feature function in the decoder that has access to the internal tree structure of each hypothesis, and hand-written rules that check if any of the following subcategorisation constraints are violated when hypotheses are combined into a new tree. These constraints need only be checked if a potentially overgeneral non-terminal is being expanded, so we do not need to check the full hypothesis tree for constraint violations. For instance, only if a non-terminal that is being expanded is the first constituent of a relative clause do we check if it contains a relative pronoun. If a constraint violation is found, we add a cost that is sufficiently high to push the hypothesis to the bottom of the hypothesis stack. We define the following constraints:

Table 1
Example of the effect of German target-side compound splitting.

Source:	Singapore is a city State
Reference:	Singapur ist ein Stadtstaat
Baseline:	Singapur ist eine Stadt des Staates
Split:	Singapur ist ein Stadtstaat

1. Relative clauses must contain a relative (or interrogative) pronoun in their first constituent (ignoring commas).
2. Modal verbs subcategorise for an infinitive, the auxiliary verbs *haben* (Engl: have) and *sein* (Engl: be) subcategorise for a past participle, or an infinitive clause (*zu* + infinitive).
3. The past participle of some verbs, mostly intransitive verbs that describe a movement or change of state, must be formed with *sein* (Engl: be) instead of *haben*, and cannot be passivised (except with the impersonal subject *es*). An example is *gestorben*, the past participle of English *die*.
4. Passive clauses, identified by the auxiliary verb *werden* or *sein* dominating a past participle, cannot subcategorise for an accusative object.[2]
5. Most subordinating conjunctions subcategorise for a finite verb, while *um* and *ohne* subcategorise for an infinitive clause. Since the treebank label for conjunctions (*KONJ*) is ambiguous, this constraint enforces the correct subcategorisation.

For verbs whose past participle is formed with *sein*, we use a list extracted from Wiktionary, manually corrected, with 260 past participle forms. All other rules are on the level of non-terminal labels, with the exception of full form lists to disambiguate the auxiliary verbs *haben*, *werden* and *sein*, which share the pre-terminal label VAFIN, but which we need to distinguish for the rules.

5.4. Compound splitting

Compositional morphology in German is productive, and a translation process that treats word forms as atomic elements is unable to account for this productive generation of new word forms. As a result, the translation of compounds may be ill-formed if they have not been observed during training. In contrast to the problems discussed previously, which were problems of overgeneralisation, this is an example where our baseline string-to-tree system shows a lack of generalisation.

At best, the failure to productively produce new compounds results in translations that remain comprehensible, e.g. producing *kulturelle Experten* instead of the compound *Kulturexperten* for the English phrase *cultural experts*. At worst, the meaning becomes distorted, as in the example sentence in Table 1, where *city State*, which corresponds to the German compound *Stadtstaat*, is instead translated as *Stadt des Staates* (Engl: *city of the state*). The translation is grammatically correct, but inaccurate. This example illustrates that enforcing grammaticality is a necessary, but not a sufficient condition for a successful translation. The model also needs to be powerful enough to generate the correct translation.

We perform compound splitting for nouns, based on a hybrid approach described by Fritzinger and Fraser (2010), using a finite-state morphology to identify compound boundaries, and frequency statistics from the corpus to choose the most probable split (Koehn and Knight, 2003). Since we perform compound splitting on the target side, we need to represent the split compounds in a way that facilitates compound merging in post-processing. For this reason, we explicitly add the junctures that join compound segments to the split representations, or a special null token if there is no juncture. We use the Zmorge morphology for German (Sennrich and Kunz, 2014), which is a variant of SMOR (Schmid et al., 2004) with an open lexicon. One advantage of Zmorge over SMOR is that Zmorge's analysis retains and marks junctures that join compound segments. We mark all junctures, including null junctures, with special characters to identify them during compound merging.

[2] This rule may misidentify the past perfect of the above-mentioned verbs of movement or change of state as passive forms. Since these verbs are all intransitive, disallowing accusative objects is still valid.

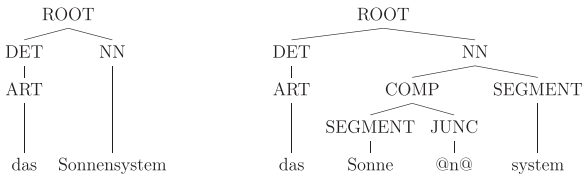

Fig. 7. Original ParZu representation without compound splitting (left) and modified version with split German compound (right).

Additionally, we represent compounds as a syntactic tree with the new pre-terminal symbols SEGMENT for noun stems and JUNC for junctures, and the non-terminal symbol COMP for compound modifiers. Minimally, compounding can be modelled with two non-terminal rules:

$$\text{NN} \rightarrow \text{COMP SEGMENT}$$
$$\text{COMP} \rightarrow (\text{COMP}) \text{ SEGMENT JUNC}$$

The second rule is recursive to allow the production of compounds with more than two segments.

While target-side identification and merging of compounds is challenging with phrase-based SMT due to its distortion model (see Stymne and Cancedda, 2011), this tree representation ensures that a (possibly empty) juncture will always be surrounded by two segments that can be merged, and that a reordering of elements is only possible if licensed by the grammar. Thus, compound merging simply consists of removing the whitespace (and special characters) around junctures. The representation of German *Sonnensystem* (Engl: solar system) before and after compound splitting is shown in Fig. 7.

A complicating factor with compound splitting is that the language model is unsuited to choose the right inflection for articles and adjectives, since the gender of the compound is determined by its last segment. We thus extend the morphological constraints discussed in Section 5.2 to split compounds by projecting the morphological feature structure of the final segment of the compound to the full compound, ignoring the morphology of compound modifiers.

6. Experiments

For SMT training and decoding, we use Moses (Koehn et al., 2007), MGIZA++ (Gao and Vogel, 2008), and KenLM (Heafield et al., 2013). We use 5-gram language models trained with SRILM (Stolcke, 2002), interpolated for minimal perplexity on newstest2012. We use training data from the ACL 2014 Ninth Workshop on Statistical Machine Translation (WMT) shared translation task, consisting of 4.5 million sentence pairs of parallel data and a total of 120 million sentences of monolingual data. We use batch MIRA (Cherry and Foster, 2012) for parameter tuning on a subset of 2000 sentences from the newstest2008–2012 test sets.[3] The WMT newstest2013 test set serves as development test set, and newstest2014 as an unseen test set.

We measure translation performance with BLEU (Papineni et al., 2002). We are aware that automatic n-gram metrics are of limited use to capture the aspect of translation that we aim to improve, namely its grammatical well-formedness. For instance, we do not expect large gains in BLEU when improving non-local agreement (such as subject-verb agreement in subordinated clauses, and verb subcategorisation), but we do expect humans to prefer the more grammatical variant. While fluency-based metrics of translation quality have been proposed (e.g. Mutton et al., 2007; Parton et al., 2011), they rely on language-specific processing, and we are not aware of such a metric for German. Also, we are wary of using syntactic parsers to measure fluency, considering that a (synchronous) parser is at the root of the translation errors that we address in this paper.

For a large-scale human evaluation, we submitted our best system to the shared translation task of the ACL 2014 Ninth Workshop on Statistical Machine Translation. In the human evaluation, it was ranked 1–2 (out of 18 systems), along with Online-B (Bojar et al., 2014). Apart from unconstrained systems, i.e. systems using additional training data, and system combinations, the next best system is a phrase-based system (PBSMT) described in Durrani et al. (2014). In a head to head comparison, our system was judged better in 57% of comparisons (ignoring ties), statistically

[3] We selected sentences shorter than 30 tokens for which a baseline SMT system produced high sentence-level BLEU score, as in (Nadejde et al., 2013).

Table 2
Size of non-terminal vocabulary and number of SCFG rules in the grammars of the SMT systems based on the syntactic annotation of various parsers. Filtered column denotes number of rules that match the source text of newstest2013.

System	Non-terminals	All rules	Hierarchical	Lexical	Filtered
Stanford Parser	165	60.7 M	39.7 M	21.0 M	8.3 M
Berkeley Parser	86	66.1 M	44.8 M	21.3 M	8.6 M
BitPar (baseline)	351	60.9 M	39.7 M	21.2 M	9.5 M
BitPar (core label set)	82	62.0 M	41.3 M	20.7 M	9.3 M
ParZu (original)	88	36.6 M	18.7 M	17.9 M	10.4 M
ParZu (modified)	125	37.6 M	19.4 M	18.1 M	10.7 M

significant at $p \leq 0.01$ according to the Sign Test. Note that the phrase-based system uses models that could also be incorporated into ours, such as generalised word representations for language modelling and various sparse feature functions.

We evaluate a second phrase-based system (referred to as *vanilla*), which is more similar to the syntactic systems in that it only uses the same 5-gram language model as the syntactic systems, whereas the system by Durrani et al. (2014) additionally uses language models on the level of morphological and POS tags, and on the level of a generalised word representation. Also, Durrani et al. (2014) use an operation sequence model and various sparse feature functions, which we did not include in the vanilla system. Finally, we use the same 2000-sentence tuning set for parameter tuning for the vanilla PBSMT and all syntactic systems, whereas Durrani et al. (2014) use 13000 sentences for tuning.

We experimented with different syntactic annotations obtained with ParZu and three other out-of-the-box German parsers: BitPar (Schmid, 2004), which is trained on the TIGER treebank (Brants et al., 2002); the Stanford Parser (Rafferty and Manning, 2008), which is trained on the NEGRA corpus (Brants et al., 1999); and the Berkeley Parser (Petrov and Klein, 2007, 2008).[4] We set up string-to-tree translation systems based on each of the different parses of the German target-language side of the parallel training data.

While BitPar, the Stanford Parser and the Berkeley Parser are all based on the NEGRA/TIGER annotation scheme, they differ in whether they include functional annotation in the non-terminal labels. The Berkeley Parser provides no functional annotation, using NP for all noun phrases, whereas BitPar and the Stanford Parser provide a functional annotation, using NP-SB for subjects, NP-OA for direct objects, among others. BitPar uses more functional categories, and for more of its constituents (including prepositional phrases and sentences), whereas the Stanford Parser only annotates subjects, accusative and dative objects. The Stanford Parser also projects the function to the pre-terminal labels, whereas the other three parsers use STTS labels (Table 2).

Even though the size of the vocabulary of target non-terminal symbols is rather divergent, the number of SCFG rules that are extracted from the corpora parsed with the NEGRA/TIGER annotation scheme differs only marginally. Coarsening the label set of BitPar to a core label set which is similar to that of the Berkeley parser even leads to a small increase in the number of rules, which we attribute to the fact that we prune singleton rules. While the NEGRA/TIGER annotation scheme annotates noun phrases as flat structures, the dependency annotation of ParZu is deeper, resulting in fewer rules being extracted due to the rule depth constraints during extraction, but more rules being applicable to the newstest2013 set.

Table 3 shows our translation results. According to BLEU, our best system outperforms the vanilla PBSMT system, but not the system by Durrani et al. (2014). However, in the human evaluation of the ACL 2014 Ninth Workshop on Statistical Machine Translation (Bojar et al., 2014), our system was judged to be significantly better than the one by Durrani et al. (2014). This finding is consistent with previous shared machine translation tasks, in which BLEU overestimated the performance of PBSMT in comparison to syntactic or rule-based systems (Callison-Burch et al., 2006; Bojar et al., 2013).

The system based on syntactic annotation obtained with BitPar is similar to the submission by Nadejde et al. (2013) to the shared machine translation task at the ACL 2013 Eighth Workshop on Statistical Machine Translation (Bojar

[4] We employed the provided German grammar for the Berkeley Parser. Unfortunately, it was not indicated in the release which treebank was used for training, but its label set follows the NEGRA/TIGER annotation scheme.

Table 3
English→German translation results on devtest (newstest2013) and test (newstest2014) sets.

System	BLEU	
	Devtest	Test
PBSMT (vanilla)	19.0	18.6
PBSMT (Durrani et al., 2014)	20.9	20.1
Stanford Parser	19.0	18.3
Berkeley Parser	19.3	18.6
BitPar (baseline)	19.5	18.6
BitPar (core label set)	19.5	18.9
ParZu	19.6	19.1
+ modified label set	19.8	19.1
+ discriminative weights for UNK	19.9	19.2
+ German compound splitting	20.0	19.8
+ syntactic constraints	20.2	20.1

et al., 2013), and we consider it our baseline. It outperforms the two setups which rely on annotation from the Stanford Parser and the Berkeley Parser.

The system based on plain ParZu syntactic annotation and without any enhancements provides slightly better translation quality (as measured in BLEU) than the BitPar baseline system. After applying our linguistically informed techniques to the ParZu translation system, we observe an overall gain of +0.7 BLEU on devtest (from 19.5 to 20.2) and +1.5 BLEU on test (from 18.6 to 20.1) over the baseline. We now discuss in more detail how the individual improvements are achieved.

Regarding the translation systems based on different syntactic annotations, we observe a difference of 0.5 BLEU on devtest, and 0.8 BLEU on test between the best and the worst system. The main factors that vary between the systems are the syntactic annotation scheme, the size of the non-terminal vocabulary, and parse quality. The relative impact of each is hard to quantify, but note the direct comparisons that differ only in one factor. The Stanford Parser, the Berkeley Parser, and BitPar use similar annotation schemes and are thus more comparable to each other than to ParZu. For the BitPar system, we test two non-terminal label sets. The original, fine-grained label set with 351 labels, and a core label set of 82 labels that is modelled after the Berkeley Parser labels. A comparison of the BitPar system (core label set) and the Berkeley Parser indicates that the former produces better trees, or at least trees more suitable for our syntactic system, with gains of 0.2–0.3 BLEU over the latter. A comparison of the two label sets for BitPar shows no difference on devtest, and a gain of 0.3 BLEU of the core label set over the original label set.

While we refrain from further attempts to quantify the effect of the non-terminal vocabulary and parsing errors on translation quality, translation errors can be traced back to both. The Berkeley Parser, the Stanford Parser and BitPar (core label set) use the same non-terminal symbol S for both main clauses and subordinated clauses. The two clauses are not interchangeable though because they differ in the position of the verb. Thus, we observe both instances where the translation system trained on Stanford parses places the verb at the last position of a main clause, and at the second position in a subordinated clause, both of which are wrong.

The original BitPar label set does make a distinction between different types of clauses, but may produce the wrong word order for another reason. In the TIGER annotation scheme, if the finite verb is a modal or auxiliary verb, the non-finite full verb and its dependents form a VP constituent. We found that BitPar produces a relatively high number of parse trees on the training text that contain a VP constituent, but no finite verb.[5] This is caused by tagging and parsing errors, for instance if the finite verb has been misanalysed as a noun or a non-finite verb. When rules learned from these parses are applied during translation, they produce sentences that are missing an auxiliary verb, or that erroneously have the verb in clause-final position, as shown in Fig. 8.

We conducted an approximate assessment of the impact of such parsing errors on translation quality in the BitPar system by identifying rules extracted from linguistically misanalysed training sentences. We looked for rules whose

[5] Specifically, we find that rule extraction extracts 13000 instances of rules whose target-side body matches NP – SB VP – OC.

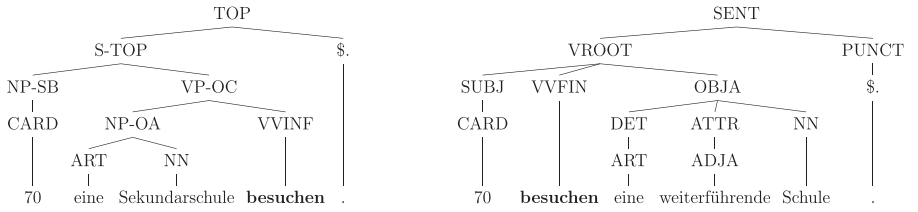

Fig. 8. Example translations of string-to-tree SMT systems trained with BitPar annotation (left) and ParZu annotation (right).

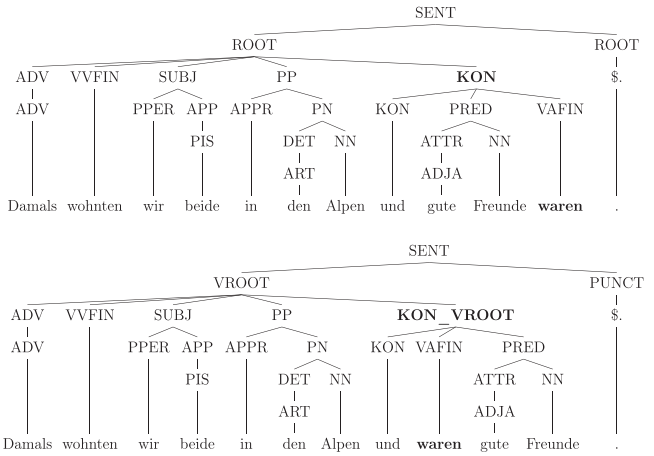

Fig. 9. Example translations of string-to-tree SMT systems trained with the original ParZu annotation (top) and with the modified version (bottom).

body either matches "NP – SB VP – OC", has the prefix "NP – SB VP – OC ." or "NP – SB VP – OC , ", has the suffix ", NP – SB VP – OC", or contains ", NP – SB VP – OC .". At least one rule matching one of these patterns has been applied by the BitPar translation system for the first-best translation of 107 out of the 3000 sentences in the newstest2013 set. This search is non-exhaustive, since it does not capture variants of the rule whose non-terminal symbols are instantiated with specific sub-derivations, for instance terminal symbols, but the number gives us a lower bound on the actual number of translations affected by parsing errors of this type. We compared the 107 affected BitPar system translations with translations of the same input sentences from the ParZu system by means of computing sentence-level BLEU scores on each of them. Taking the difference of the sentence-level BLEU scores of the ParZu system output and of the BitPar system output, we found that the 107 ParZu system translations are on average +0.45 absolute better than the BitPar system translations with respect to sentence-level BLEU. The average difference over the whole newstest2013 set is at just +0.06 absolute and thus much lower. The high average advantage of the ParZu system over the BitPar system on translations with application of rules extracted from training instances with defective BitPar syntactic parses highlights the relevance of a linguistically sound annotation.

Even though the system trained on ParZu parses performs best, it still suffers from overgeneralising non-terminal symbols. We obtain an improvement of 0.2 BLEU on devtest from modifying the ParZu label set to distinguish between different types of unattached words, between different types of coordinations, and between prenominal and postnominal genitive modifiers. Fig. 9 illustrates a grammatical error originating from the overgeneral label KON. The sentence is a coordination of two main clauses, but because the label KON is used for coordinations both within main clauses and subordinated clauses, the system trained on the original label set is allowed to use rules learned from either, and erroneously chooses verb-last word order for the coordinated element. With the modified label set, only rules learned from main clause coordinations are considered, i.e. those with the head symbol KON_VROOT, and the correct word order is chosen.

Discriminatively learning the best labels for unknown words, as described in Section 3.2, gives an improvement of 0.1 BLEU on devtest.[6] Table 4 shows how BLEU is affected for sentences with and without unknown words. The

[6] Note that we also essentially disable glue rules in this step, setting their weight to −100.

Table 4
English→German evaluation of labelling strategies for unknown words (on newstest2013).

BLEU				
Subset	Sentences	Baseline	Discriminative	Manual
All	3000	19.8	19.9	19.8
With unknown word	527	20.7	21.1	20.6
Without unknown word	2473	19.3	19.4	19.4

Table 5
English→German evaluation of target-side compound splitting (on newstest2013). Reference and test output are tokenised, but with compounds merged.

BLEU			
Subset	Sentences	Baseline	Split
All	3000	19.9	20.0
With compound	511	17.1	17.7
Without compound	2489	20.5	20.5

performance gain from the discriminative training strategy comes mostly from sentences that contain unknown words ($n = 527$; BLEU +0.4). We find that the label distribution learned from singletons is so biased towards the label NN in German that all unknown words are assigned the label NN in the baseline system. With discriminatively learned costs, the label distribution is flatter, but still noisy. In order to determine if better prediction of labels for unknown words, e.g. by using information from the source side, could lead to further score improvements, we performed an oracle experiment in which we manually assigned pre-terminal symbols to all 480 unknown words in newstest2013. The most frequent labels are named entity (NE; 65%), noun (NN; 14%), adjective (ADJA; 13%) and number (CARD; 4%).[7] Somewhat surprisingly, this oracle experiment performs worse than discriminative training. One advantage of discriminative training over the manual labels is that it is not restricted to pre-terminal symbols. Among the most frequent labels assigned to unknown words are the non-terminal symbols for appositions (APP; 12%), attributes (ATTR; 12%) and subjects (SUBJ; 7%). If we use discriminative weights, but restrict the set of possible symbols for unknown words to pre-terminal symbols, the majority of the performance gain over the oracle system is neutralised. On average, the effect of compound splitting is slightly positive. Table 5 shows results of compound splitting for sentences with at least one split compound in the output of the experimental system, and for sentences with none.[8] While we observe an improvement in BLEU by 0.6 points for those 17% of sentences where new compounds have been produced, performance for other sentences remains stable. We used the same log-linear weights for both systems, optimised on the system without compound splitting.

The application of syntactic constraints, both those relating to noun phrases (Section 5.2), and those relating to verb subcategorisation (Section 5.3) resulted in an average improvement of 0.2 BLEU on devtest. Table 6 shows how often the syntactic constraints are violated in our 3000-sentence test set, and how enabling the constraints affects BLEU. The statistics are on a per-sentence level, and a sentence may violate multiple constraints. Of the 60% of sentences whose baseline 1-best translation does not violate any constraint, about 7% differ due to pruning effects. The effect on BLEU is slightly negative. Noun phrase agreement violations are the most frequent, being triggered in one third of the sentences. For this subset, enforcing noun phrase agreement results in a 0.6 BLEU point improvement over the system without constraints. The verb subcategorisation constraints are violated less frequently, in about 10% of sentences, and have a smaller effect on BLEU. We observe an improvement of 0.3 BLEU on the translations that violate a verb subcategorisation constraint.

[7] This distribution indicates that our baseline, the label distribution of singletons in the target text, is no good model for labelling unknown source words. NN is the predominant label for singletons in German (51%) due to its compounding nature, and we are lucky that the actually predominant type of unknown words, names, shows similar syntactic properties.

[8] We only split rare compounds in the training data (frequency <5), so frequent compounds are translated as an atomic unit.

Table 6
English→German evaluation of syntactic constraints (on newstest2013), showing number of sentences that violate a constraint, and BLEU score without and with enforcing of constraints.

BLEU			
Subset	Sentences	Baseline	Constrained
All	3000	20.0	20.2
No constraint violation	1785	22.4	22.3
Noun phrase agreement	1034	17.6	18.2
Verb subcategorisation	316	16.1	16.4

Table 7
Example translation without and with enforcing of syntactic constraint.

Source:	Since then, the mirrors in optical telescopes **have become** increasingly large [...]
Reference:	Seitdem **wurden** die Spiegel der optischen Teleskope immer größer [...]
Baseline:	Seitdem **haben** die Spiegel in optischen Teleskope immer groß **geworden** [...]
Constrained:	Seitdem **ist** der Spiegel in optische Teleskope immer groß **geworden** [...]

In a manual analysis, we found that the majority of sentences whose baseline translation violates a constraint were improved by enforcing the constraint. A minority of translations did not improve. For 7% of the sentences that violate a constraint, enforcing the constraint has no visible effect on the translation. A translation may be correct even if the derivation is grammatically unsound, and the SCFG may choose a different derivation with the same translation output if constraints are enforced.

We found isolated cases of false positives, and estimate their rate to be under 5%. For instance, nouns of measure typically remain uninflected even if they have a plural meaning. The sentence *40 Prozent sind infiziert* (Engl. *40 percent are infected*) is grammatically correct, but violates our subject–verb agreement constraint. Refining the constraints could further reduce the false positive rate.

In some cases, a constraint violation is correctly identified, but the system is either unable to produce a correct derivation, or another erroneous, but higher-scoring derivation is selected instead. Table 7 shows an example where the syntactic constraints prevent the erroneous verb complex *haben geworden*, leading the model to produce the grammatically correct *ist geworden* instead. However, the constrained system is less accurate, putting the subject and verb in the singular.

In summary, our experiments provide both empirical support that the overgeneralisation phenomena that we discussed in the previous section degrade translation quality, and show that this problem can be mitigated through choosing a suitable syntactic representation, refining overgeneral symbols in the syntactic label set, and adding linguistically motivated constraints. Eliminating ill-formed derivations can only improve translation quality if the model is able to produce derivations that are both well-formed and accurate. To increase the generation power of our model, we applied target-side compound splitting, and found that it successfully generated compounds that our baseline system could not produce.

7. Related work

Various authors have focused on data sparseness that stems from syntactic constraints during rule extraction and/or decoding, and have proposed methods to relax syntactic constraints (Venugopal and Zollmann, 2006; Chiang, 2010; Hoang, 2011; Hanneman and Lavie, 2011; Burkett and Klein, 2012). These relaxations techniques increase the likelihood of being able to produce full trees during parsing, but at the cost of potentially allowing more overgeneralisations. For instance, Hanneman and Lavie (2011) coarsen the label set by collapsing labels in one language based on their alignment probabilities to the labels in the other language. Such an approach can help to reduce rule sparseness, especially in a tree-to-tree setting where the joint label set is much larger than in a string-to-tree system, but can also remove meaningful distinctions from the label set. For example, the distinction between subjects and objects in German, which

is morphologically marked, could be lost if the labels are collapsed because they are aligned to the label NP in the other language. Our research goes the opposite route of adding new syntactic constraints.

Our sparse features for syntactic models are similar to those investigated in Chiang et al. (2009), Chiang (2010). Allowing soft matches during non-terminal expansion was proposed by Chiang (2010) as a way of relaxing the matching constraint in a tree-to-tree system with a large joint label set. We use this idea to model the label for unknown words instead.

To improve morphological agreement, Koehn and Hoang (2007) add morphological information as an additional factor in the training corpus, and use 7-gram language models trained on this morphological level in addition to a surface form language model. Fraser et al. (2012) model inflection in phrase-based SMT by representing the German side through word stems and morphological markup rather than full word forms. After the main decoding step that produces this underspecified German representation, they predict the final inflection with CRF sequence models, using the stem and morphological markup as features. Weller et al. (2013) extend this work by including source-side subcategorisation knowledge into the inflection prediction model. All methods rely on sequence models, which are unsuitable to model discontiguous phenomena such as subject-verb agreement in German subordinated clauses, or the agreement between the head of a split compound, i.e. its last element, and its determiner and attributes.

A related method to enforce morphosyntactic constraints such as morphological agreement is to implement this as a postprocessing step, in which the original translation hypothesis is linguistically analysed, and rules on the basis of this analysis are engineered to identify and fix syntactic errors (Stymne and Ahrenberg, 2010; Rosa et al., 2012). A challenge of such a post-processing approach is that tools for linguistic analysis may perform poorly on ill-formed SMT output. Modelling syntactic constraints at decoding time has the advantage that we do not need to adapt any linguistic models to SMT output. Also, we do not need to explicitly model the error correction, which is non-trivial. For instance, if a translation hypothesis violates subject–verb agreement, we cannot reliably decide on the basis of the translation hypothesis alone if the inflection of the subject should be changed to match that of the verb, or vice-versa. Instead, we let the SMT model produce and select alternative translations.

Our approach is also comparable to previous work on using syntactic information as features to evaluate or rerank translations. While we believe that these results may be parser-specific, past experiments found that including parsing probability as a feature did not help SMT performance (Och et al., 2004; Post and Gildea, 2008). Discriminative models that extract features from parsing output have been more successful (Collins et al., 2005; Mutton et al., 2007; Cherry and Quirk, 2008; Carter and Monz, 2010). Compared to a reranking approach, our syntactic constraints are applied early during search, which means that we are not in danger of filling the *n*-best list with translations that all violate a constraint. Also, we expect features extracted from the parse tree to help little if the parser yields the same structure for grammatical and ungrammatical sentences. Foster (2007) explores parser performance on an artificially created ungrammatical treebank, and found that "[agreement errors do] not generally distract [the Bikel parser and the Charniak and Johnson parser] from finding the correct analysis". While ignoring agreement errors may be a desirable property for parser robustness, it limits the usefulness of parsers for fixing agreement errors through reranking.

Stymne (2009) and Stymne and Cancedda (2011) discuss compound splitting on the target side. The added difficulty compared to source-side compound splitting is that compounds need to be merged again in a post-processing step. In phrase-based and (unlabelled) hierarchical SMT models, reorderings during translation can produce the wrong word order, at worst making compound segments discontiguous. This makes the process of identifying and merging compounds challenging. Stymne and Cancedda (2011) use sequence labelling models that are based on lexical features and part-of-speech tags, both to improve word order and to identify and merge compounds. Our syntactic annotation ensures that no reorderings of segments outside of the compound are possible, making compound merging trivial.

8. Conclusions

Out-of-the box parsers and treebanks do not necessarily provide suitable syntactic annotation for string-to-tree statistical machine translation. Focusing on German as a target language, we have discussed various ways in which string-to-tree translation models overgeneralise. Earlier research has focused more on how the syntactic representation overconstrains rule extraction and decoding in syntax-based translation. This paper highlights another important aspect, namely the fact that grammars learned from treebanks may impose too few constraints on tree derivation, and thus fail to avoid ill-formed output.

Linguistically, most of the grammatical errors that we identified are morphological, as is typical for translation into morphologically rich languages, but we have also identified overgeneralisations that cause word order errors.

We have discussed how the syntactic representation underlying a string-to-tree model affects translation quality, and have shown novel techniques to introduce additional linguistic knowledge into a string-to-tree translation system, including morphological agreement and subcategorisation constraints for noun phrases and verbs, a syntactic representation for target-side compound splitting and merging, and discriminative learning of labels for unknown words. We submitted the system described in this article to the ACL 2014 Ninth Workshop on Statistical Machine Translation, where it was ranked 1–2 (out of 18) and significantly outperformed state-of-the-art phrase-based systems.

As future work, we plan to further refine our syntactic constraints for phenomena such as coordinations. Furthermore, we could extend our work by adding more subcategorisation constraints, for instance specifically modelling possible subcategorisation frames of full verbs. Grammatical well-formedness is a necessary, but not a sufficient condition of a good translation. Thus, we also plan to increase the expressive power of our models by learning to produce inflectional forms that do not occur in the training corpus.

Acknowledgements

The research leading to these results has received funding from the Swiss National Science Foundation under Grant P2ZHP1_148717, and from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

References

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., Specia, L.,August 2013. Findings of the 2013 workshop on statistical machine translation. In: Proceedings of the Eighth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Sofia, Bulgaria, pp. 1–44.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., Tamchyna, A.,June 2014. Findings of the 2014 workshop on statistical machine translation. In: Proceedings of the Ninth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Baltimore, MD, USA, pp. 12–58.

Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G., 2002. The TIGER treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol.

Brants, T., Skut, W., Uszkoreit, H., 1999. Syntactic annotation of a German newspaper corpus. In: Proceedings of the ATALA Treebank Workshop, Paris, France, pp. 69–76.

Burkett, D., Klein, D.,July 2012. Transforming trees to improve syntactic convergence. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Jeju Island, Korea, pp. 863–872.

Callison-Burch, C., Osborne, M., Koehn, P., 2006. Re-evaluating the role of Bleu in machine translation research. In: Proceedings the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, pp. 249–256.

Carter, S., Monz, C., 2010. Discriminative syntactic reranking for statistical machine translation. In: The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)., Denver, Colorado, USA.

Cherry, C., Foster, G.,2012. Batch tuning strategies for statistical machine translation. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL HLT '12. Association for Computational Linguistics, Montreal, CA, pp. 427–436.

Cherry, C., Quirk, C., 2008. Discriminative, syntactic language modeling through latent SVMs. In: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas.

Chiang, D.,2005. A hierarchical phrase-based model for statistical machine translation. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, USA, pp. 263–270.

Chiang, D., 2007. Hierarchical phrase-based translation. Comput. Linguist. 33 (2), 201–228.

Chiang, D.,2010. Learning to translate with source and target syntax. In: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Uppsala, Sweden, pp. 1443–1452.

Chiang, D., Knight, K., Wang, W.,2009. 11,001 new features for statistical machine translation. In: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Boulder, CO, pp. 218–226.

Collins, M., Roark, B., Saraclar, M., 2005. Discriminative syntactic language modeling for speech recognition. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, pp. 507–514.

Durrani, N., Haddow, B., Koehn, P., Heafield, K.,June 2014. Edinburgh's phrase-based machine translation systems for WMT-14. In: Proceedings of the Ninth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Baltimore, MD, USA, pp. 97–104.

Foster, J., 2007. Treebanks gone bad: parser evaluation and retraining using a treebank of ungrammatical sentences. Int. J. Doc. Anal. Recogn. 10 (3), 129–145.

Foth, K.A., 2005. Eine umfassende Constraint-Dependenz-Grammatik des Deutschen. University of Hamburg, Hamburg.
Fraser, A., Weller, M., Cahill, A., Cap, F.,2012. Modeling inflection and word-formation in SMT. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Avignon, France, pp. 664–674.
Fritzinger, F., Fraser, A.,2010. How to avoid burning ducks: combining linguistic analysis and corpus statistics for German compound processing. In: Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR. WMT '10. Association for Computational Linguistics, Uppsala, Sweden, pp. 224–234.
Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., Thayer, I.,2006. Scalable inference and training of context-rich syntactic translation models. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, USA, pp. 961–968.
Galley, M., Hopkins, M., Knight, K., Marcu, D., 2004. What's in a translation rule? In: HLT-NAACL '04.
Gao, Q., Vogel, S.,2008. Parallel implementations of word alignment tool. In: Software Engineering, Testing, and Quality Assurance for Natural Language Processing. Association for Computational Linguistics, Columbus, OH, pp. 49–57.
Hanneman, G., Burroughs, M., Lavie, A., June 2011. A general-purpose rule extractor for SCFG-based machine translation. In: Proceedings of Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation, Portland, Oregon, USA, pp. 135–144.
Hanneman, G., Lavie, A.,2011. Automatic category label coarsening for syntax-based machine translation. In: Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation. SSST-5. Association for Computational Linguistics, Portland, OR, pp. 98–106.
Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P., August 2013. Scalable modified Kneser–Ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, pp. 690–696.
Hoang, H., 2011. Improving Statistical Machine Translation with Linguistic Information. University of Edinburgh, Ph.D. thesis.
Hopkins, M., Langmead, G., 2010. SCFG decoding without binarization. In: EMNLP, pp. 646–655.
Hopkins, M., Langmead, G., Vo, T.,July 2011. Extraction programs: a unified approach to translation rule extraction. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, Scotland, pp. 523–532.
Koehn, P., Hoang, H.,2007. Factored translation models. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Association for Computational Linguistics, Prague, Czech Republic, pp. 868–876.
Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.,2007. Moses: open source toolkit for statistical machine translation. In: Proceedings of the ACL-2007 Demo and Poster Sessions. Association for Computational Linguistics, Prague, Czech Republic, pp. 177–180.
Koehn, P., Knight, K.,2003. Empirical methods for compound splitting. In: EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Budapest, Hungary, pp. 187–193.
Koehn, P., Och, F.J., Marcu, D., 2003. Statistical phrase-based translation. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pp. 48–54.
Kübler, S., September 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In: Proceedings of RANLP, Borovets, Bulgaria, pp. 293–300.
Kübler, S.,2008. The PaGe 2008 shared task on parsing German. In: Proceedings of the Workshop on Parsing German. Association for Computational Linguistics, Columbus, OH, pp. 55–63.
Marcu, D., Wang, W., Echihabi, A., Knight, K., 2006. SPMT: statistical machine translation with syntactified target language phrases. In: EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, pp. 44–52.
Mutton, A., Dras, M., Wan, S., Dale, R., 2007. GLEU: automatic evaluation of sentence-level fluency. In: Carroll, J.A., van den Bosch, A., Zaenen, A. (Eds.), Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. The Association for Computational Linguistics, pp. 344–351.
Nadejde, M., Williams, P., Koehn, P.,August 2013. Edinburgh's syntax-based machine translation systems. In: Proceedings of the Eighth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Sofia, Bulgaria, pp. 170–176.
Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D., 2007. The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932.
Och, F.J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., Radev, D.,2004. A smorgasbord of features for statistical machine translation. In: HLT-NAACL 2004: Main Proceedings. Association for Computational Linguistics, Boston, MA, USA, pp. 161–168.
Och, F.J., Ney, H.,2002. Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02. Association for Computational Linguistics, Morristown, NJ, USA, pp. 295–302.
Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.,2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Philadelphia, PA, pp. 311–318.
Parton, K., Tetreault, J., Madnani, N., Chodorow, M.,July 2011. E-rating Machine Translation. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, Scotland, pp. 108–115.
Petrov, S., Klein, D., April 2007. Improved Inference for Unlexicalized Parsing. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, Rochester, New York, pp. 404–411.
Petrov, S., Klein, D., June 2008. Parsing German with latent variable grammars. In: Proceedings of the Workshop on Parsing German at ACL '08, Columbus, OH, pp. 33–39.
Post, M., Gildea, D., 2008. Parsers as language models for statistical machine translation. In: Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas.

Rafferty, A.N., Manning, C.D., June 2008. Parsing three German treebanks: lexicalized and unlexicalized baselines. In: Proceedings of the Workshop on Parsing German at ACL '08, Columbus, OH, pp. 40–46.

Rosa, R., Mareček, D., Dušek, O.,2012. DEPFIX: a system for automatic correction of Czech MT outputs. In: Proceedings of the Seventh Workshop on Statistical Machine Translation. WMT '12. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 362–368.

Schiller, A., Teufel, S., Stöckert, C., Thielen, C., 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Tech. rep. Institut fur maschinelle Sprachverarbeitung, Stuttgart.

Schmid, H.,2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In: Proceedings of the 20th International Conference on Computational Linguistics. COLING '04. Association for Computational Linguistics, Geneva, Switzerland.

Schmid, H., Fitschen, A., Heid, U., 2004. A German computational morphology covering derivation, composition, and inflection. In: Proceedings of the IVth International Conference on Language Resources and Evaluation (LREC 2004), pp. 1263–1266.

Sennrich, R., Kunz, B., May 2014. Zmorge: a German morphological lexicon extracted from wiktionary. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland.

Sennrich, R., Volk, M., Schneider, G., 2013. Exploiting synergies between open resources for German dependency parsing, POS-tagging, and morphological analysis. In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2013, Hissar, Bulgaria, pp. 601–609.

Shieber, S.M., Schabes, Y.,1990. Synchronous tree-adjoining grammars. In: Proceedings of the 13th Conference on Computational Linguistics - Volume 3. COLING '90. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 253–258.

Stolcke, A., 2002. SRILM – an extensible language modeling toolkit. In: Seventh International Conference on Spoken Language Processing, Denver, CO, pp. 901–904.

Stymne, S., 2009. A comparison of merging strategies for translation of German compounds. In: Lascarides, A., Gardent, C., Nivre, J. (Eds.), EACL (Student Research Workshop). The Association for Computer Linguistics, pp. 61–69.

Stymne, S., Ahrenberg, L., 2010. Using a grammar checker for evaluation and postprocessing of statistical machine translation. In: Chair), N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (Eds.), Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC '10). Valletta, Malta, pp. 2175–2181.

Stymne, S., Cancedda, N.,2011. Productive generation of compound words in statistical machine translation. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. WMT '11. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 250–260.

Telljohann, H., Hinrichs, E.W., Kübler, S., 2004. The TüBa-D/Z treebank: annotating German with a context-free backbone. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, Lisbon, Portugal, pp. 2229–2235.

Venugopal, A., Zollmann, A.,2006. Syntax augmented machine translation via chart parsing. In: Proceedings on the Workshop on Statistical Machine Translation. Association for Computational Linguistics, New York City, pp. 138–141.

Versley, Y., 2005. Parser evaluation across text types. In: Fourth Workshop on Treebanks and Linguistic Theories (TLT), Barcelona, Spain.

Weller, M., Fraser, A., Schulte im Walde, S.,August 2013. Using subcategorization knowledge to improve case prediction for translation to German. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Sofia, Bulgaria, pp. 593–603.

Williams, P., Koehn, P.,July 2011. Agreement constraints for statistical machine translation into German. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, Scotland, pp. 217–226.

Williams, P., Koehn, P.,June 2012. GHKM rule extraction and scope-3 parsing in Moses. In: Proceedings of the Seventh Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montréal, CA, pp. 388–394.